440-TP-003-001

Science Software Data Server Access: A Trade-Off Study Analysis

Technical Paper—Not intended for formal review or government approval.

Technical Paper

28 February 1995

Prepared Under Contract NAS5-60000

RESPONSIBLE ENGINEER

Mac McDonald for Chhaya Rao	2/28/95
Chhaya S. Rao, PDPS Scientist/Engineer EOSDIS Core System Project	Date

SUBMITTED BY

Parag N. Ambardekar	2/28/95
Parag Ambardekar, PDPS Manager EOSDIS Core System Project	Date

Hughes Applied Information Systems
Landover, Maryland

This page intentionally left blank.

Abstract

The issues, analysis and results of the Science Software Data Server Access trade study is documented in this Technical paper. Science software may not always be able to identify its input prior to execution. It may need direct access to Data Servers or an independent database in order to search and obtain relevant data prior to execution. This study analyzes MODIS, CERES, MISR and ASTER instruments and presents reasons why pre-staged input data cannot be used for these cases. The nature of the information searched and the potential volume of the retrieved data inputs are also presented. The different trade-off alternatives namely: 1) to limit Data Server access to the staging process, 2) to allow all potential data and relevant metadata to reside on a separate dedicated database (to be accessed during software execution), 3) to allow science software to access Data Servers directly (queries, data retrievals), and 4) to allow science software to access Data Servers through the SDP Toolkit (queries, data retrievals) are explored.

Keywords: science software, data server access, staging, pre-staging, ASTER, MODIS, CERES, MISR, database, DBMS

This page intentionally left blank.

Contents

Abstract

1. Introduction

2. Executive Summary

3. Major Trade Alternatives

4. Analysis Approach

5. Selection Criteria

6. Evaluations

7. Recommendations and Conclusions

Abbreviations and Acronyms

1. Introduction

1.1 Purpose and Scope

This white paper documents the results of the Science Software Data Server Access trade study which was conducted to determine the need of EOS science software to have direct access to a data server/DBMS during execution. The white paper addresses the issue itself, describes the analysis performed during the trade study and presents the results and conclusions drawn.

1.2 Trade Description

Science software may not always be able to identify its input data prior to execution. It may need direct access (while executing) to data servers or an independent database in order to search for appropriate data and, when appropriate data is identified, to obtain that data. But specific requirements in support of this type of access needed to be identified by the science software developers.

The Science Software Data Server Access trade study was conducted to determine the need of EOS (MODIS, CERES, MISR and ASTER) science software to have direct access to a data server / DBMS during execution. Currently, the system is designed to isolate science software access to the ECS system. This is done by providing the SDP Toolkit to buffer all science software interaction with the system and by pre-staging all input data required by the science software prior to execution. This minimizes contention for data server resources during production and assumes that the input data can be identified prior to execution. However, there may be some cases, where the input data to be accessed may be known only during the course of execution, thereby requiring direct access to the data server or an independent DBMS. The objective of this trade study was to identify these cases and determine how they would be supported.

Some of the major alternatives explored were 1) to limit data server access to the staging process (as originally proposed), 2) to allow all potential data and relevant metadata (to be accessed during software execution) to reside on a separate, dedicated database, 3) to allow science software to access data servers directly (queries, data retrievals), and 4) to allow science software to access data servers through the SDP Toolkit (queries, data retrievals).

1.3 Organization

The paper is organized as follows:

- Section 1 presents the purpose and scope of the white paper, a description of the trade study, the organization of the white paper and the logistics concerning its review.
- Section 2 provides an executive overview with emphasis on major conclusions drawn or major directions taken.

- Section 3 presents the major options / trade alternatives available along with their high level descriptions.
- Section 4 describes the analysis approach used during the trade study.
- Section 5 lists the selection criteria considered when selecting a trade alternative.
- Section 6 records the evaluation data compiled during the comparative study of the different trade alternatives and presents conclusions drawn.
- Section 7 presents the final recommendations made .
- The Appendices provide information on the abbreviations and acronyms used.

1.4 Acknowledgements

The contributions of the members of the ASTER, CERES, MISR and MODIS science teams and the ARC and EDS support teams are sincerely appreciated.

1.5 Review and Approval

Questions regarding technical information contained within this paper should be addressed to the following ECS and/or GSFC contacts:

- ECS Contact

  Mac McDonald, PDPS System Engineer

  (301)-925-0364

  mac@eos.hitc.com

- GSFC Contact

  Steve Kempler, PDPS Manager

  (301)-286-7766

  steven.j.kempler@gsfc.nasa.gov

Questions concerning distribution or control of this document should be addressed to:

Data Management Office
The ECS Project Office
Hughes Applied Information Systems
1616 McCormick Dr.
Landover, MD 20785

1.6 Applicable and Reference Documents

MA9401V1 DBMS Evaluations: Technical white paper, MA9401V1

2. Executive Summary

The Science Software Data Server Access trade study working paper identifies the EOS (ASTER, CERES, MODIS and MISR) science software that require access to a data server / DBMS during software execution and presents reasons why pre-staged input data cannot be used for these cases. In addition, for these cases, it identifies the nature of the information searches and the potential volume of the retrieved data inputs. The working paper presents the pros / cons of the different trade alternatives studied and provides the conclusions drawn regarding how these requirements can be supported. A summary of the results of this trade study is as follows:

(A) ASTER

It was determined that ASTER software required direct access to Lookup tables (LUTs) during execution. The LUTs (used during atmospheric correction of ASTER radiance data) are approximately 9 - 10 gigabytes. A query to this table is based on 7 parameters for each band and file version, 4 of which can be determined during execution only. Several hundred queries (average of about 1 access every second (with a worst case of 40,000 accesses per scene)) need to be made during execution. During the day, the entire LUT would, more or less, have to be accessed.

Since allowing direct access to the data server would have many performance impacts, this approach is being discouraged. The other alternatives studied were to either (a) store the LUTs as flat files and have them pre-staged or (b) provide a separate, independent database dedicated to ASTER processing. The working paper addresses the pros/cons of each option. The option of providing a separate database (preferably a relational database which is suited to ASTER processing) was selected and the estimated cost of this option is currently being worked with ESDIS.

(B) MODIS

MODIS input data access needs are still being studied. A related study is currently under way to (a) develop probable MODIS data processing scenarios based on inputs from the MODIS science teams and, (b) determine how these scenarios can be accommodated within the Planning/Processing environment. The MODIS direct data access needs will become more apparent as a result of this related study and at that time (during the CDR time frame) a better determination can be made regarding how these requirements should be supported.

(C) CERES

No direct data server access required during execution of CERES software related to ancillary and calibration coefficient data. The ancillary files (approx. 1.6 Mbytes) and calibration coefficient files (1.82 Mbytes, maximum) can be pre-staged prior to execution (part of existing basic design baseline) and these files can be searched effectively during execution without any need for a separate DBMS.

(D) MISR

No requirements for dynamic access to input data have been identified to date. Confirmation of this can be made once the MISR system design has matured.

ANALYSIS PLAN

Plan to continue working with the algorithm/science software development teams in determining whether any additional direct data access requirements (during execution) need to be identified.

3. Major Trade Alternatives

3.1 Major Alternatives Studied

The following major alternatives were explored for each case where the science software required access to input data during execution:

- Limit data server access to the staging process (part of basic design baseline already).
- Allow all potential data and relevant metadata (to be accessed during software execution) to reside on a separate, dedicated database.
- Allow science software to access data servers directly (queries, data retrievals).
- Allow science software to access data servers through the SDP Toolkit (queries, data retrievals).

Regarding the last two bullets, data server access by science software presents many performance concerns. Searching for data could contest with users for query satisfaction. Data retrieval could contest with users and other data production activities for data server resources. Both of these cases may cause the current process to pause for a significant amount of time in its processing stream. This will require a decision to either not effectively utilize the processing resource (if no other process is allowed to share processing time) or to "swap out" that process until the query or data retrieval are completed. The swap out solution is likely to require that additional staging and temporary storage space be made available.

Allowing all potential input data and metadata (to be accessed during execution) to reside on a separate, dedicated database might avoid search conflicts with other users. But this alternative must be evaluated against the number of searches to be performed and the complexity of maintaining a database.

Pre-staging of all potential data inputs as flat files becomes a potential matter of extra disk space requirements and requires accesses to the data server for retrieval of some amount of data that is never used.

This page intentionally left blank.

4. Analysis Approach

4.1 Analysis Techniques

The following analysis techniques were used during the course of this trade study:

For each science software studied, convincing arguments concerning the need to directly access the input data during execution had to be established. To do this the following questions were asked concerning each science software considered:

- Define the requirements regarding science software direct access to the data server/DBMS during execution; indicate why the input data should not/cannot be pre- staged (as flat files) prior to execution.
- Characterize the nature and frequency of information searches in order to estimate the effect on database performance,
- Determine the potential volume of data inputs to evaluate disk sizing impacts and performance impacts for retrieval purposes.

Once the need to have direct access to certain input data during execution was established, a comparative study of the different trade alternatives was made to determine which alternative was most suitable. The trade alternatives considered were: (a) pre-stage data as flat files (b) allow the accessed data to reside on a separate, independent database (c) access the data server directly or via SDP Toolkit. The suitability, cost and performance impacts of each option were considered in addition to the other selection factors listed in Section 5. Based on the outcome of this study, a recommendation was made concerning how the direct data access requirements of the science software could be supported.

This page intentionally left blank.

5. Selection Criteria

5.1 Evaluation Selection Criteria

The following selection factors were considered when selecting a suitable approach for supporting each science software's direct data access requirements:

- Performance Impacts
- Cost
- Risk in implementing
- Maturity/Stability of alternative design
- Maintainability
- Evolvability

This page intentionally left blank.

6. Evaluations

6.1 Dynamic Access to Input Data Requirements

Meetings with the different science software teams have established the fact that certain science software might require access to a data server or a DBMS during execution. In many cases, since the parameters needed for the retrieval of such data are 'dynamically' determined during execution only, they cannot be specified in advance to allow pre-staging of the data to be done. The dynamic data access needs of MODIS, CERES, MISR and ASTER are given below.

6.1.1 MODIS

One of the MODIS software subsystems [supporting MODIS data processing] requires dynamic access to input data that can be identified during execution only. This package dynamically generates PGE process control script files. Scheduling of the processing jobs can be accomplished by utilizing a relational database (that contains Main and Process Control tables). Subsequent steps in the processing can then be initiated based on inputs updated during the previous processing steps.

To better understand the MODIS data access needs, effort is currently under way to (a) develop MODIS data processing scenarios based on inputs from the MODIS science teams and, (b) determine how these scenarios can be accommodated within the Planning / Processing environment. The MODIS direct data access requirements should become more apparent as a result of this study and this will allow a better determination to be made regarding how these requirements need to be supported.

6.1.2 CERES (TRMM)

CERES uses ancillary input data files during processing. These ancillary data files include the CERES Angular Directional Models (CADMs). The CADMs will be stored as a function of scene type and several geometry angles. In the past, these models were stored as a flat file and each program that needed to access them had to calculate pointers, offsets and know the format. A full set of CADMs are approximately 1.6 Mbytes and contain 3 types of files: LW, WN and SW CADM files. LW and WN CADM files are accessed approximately 5.9 million times to process one day of data for one instrument. SW CADM file is accessed about 3 million times to process one day of data for one instrument. For each footprint, approx. 6,790 bytes could be retrieved from the 3 CADM files.

A database is not needed to store the ancillary files though it would be easier to access (via a set of indexes that would return an array of values). All the CADM files (1.6Mbytes worth) could be pre-staged prior to execution.

CERES also uses calibration coefficient files. For calibration coefficients files, (for most days) a subset of the files could be identified (and be staged) before execution . For calibration days, all the calibration coefficient files (corresponding to 1.82 mbytes) would have to be staged (about every two weeks).

data base is not needed to store the calibration coefficients.

The ERBE Heritage software uses an INFORMIX database management system to help submit and track ERBE nonscanner processing runs by tracking the run-time parameters for output products. As a front end process, it helps in setting up processing by identifying parameters such as number of records, mode of scan, etc. After the processing, it keeps track of output parameters such as the time it ran and also critical information from quality control files for science validation.

For CERES, there are no non-scanners, so this is not an issue.

*Conclusion: No direct data server access required during execution of CERES software related to ancillary and calibration coefficient data. The ancillary files (approx. 1.6 Mbytes) and calibration coefficient files (1.82 mbytes, maximum) can be staged prior to execution (part of existing basic design baseline) and these files can be searched effectively dur*ing execution without any need for a separate DBMS .

6.1.3 MISR

No requirements for dynamic access to input data have been identified to date. Confirmation of this can be made once the MISR system design has matured.

6.1.4 ASTER

ASTER science software needs to dynamically access data from a ten-gigbyte lookup table (LUT) that is used during the atmospheric correction of ASTER radiance data. The most fundamental requirement, with regards to the atmospheric correction LUT, is to obtain an atmospheric correction value based on values for a set of independent parameters. Several hundred accesses will be necessary per ASTER scene per band (i.e. worst case of 40,000 accesses per scene, however, some sort of caching scheme can probably be employed to drastically reduce the number of accesses per day). During the course of a day's worth of processing it can be expected that essentially the entire table will need to be available.

There are two basic ways to implement this requirement. The first is via a data base. During execution, ASTER code would determine the independent parameters needed for LUT data retrieval and would then retrieve the LUT data directly from the data base. Alternately, the LUT could be split into many files. All these files would be pre-staged prior to execution and the ASTER software would determine which files to read after calculating the values of the parameters during execution.

The LUTs are needed for the creation of the Level 2 Surface Radiance / Reflectance Products for the VNIR and SWIR channels. These products will be requested via ON-Demand production requests.

6.1.4.1 The ASTER Lookup Table

The ASTER Lookup table (LUT) used during atmospheric correction of ASTER radiance data is approximately 9 - 10 gigabytes in size. A lookup 'query' to the LUT returns an atmospheric correction record that is retrieved based on the following attributes:

1. File version #
2. ASTER band number
3. Molecular Optical Depth
4. Aerosol Optical Depth
5. Aerosol Single Scatter Albedo
6. Aerosol Size Distribution
7. Solar Zenith Angle
8. Relative Azimuth Angle
9. View Zenith Angle

Of the parameters mentioned above, four are obtained "dynamically" from MISR, MODIS or NMC data. Therefore, the precise section of the LUT required for a given correction cannot be known in advance of staging time.

A record in the LUT will be a block of floating point numbers (probably less than 50; around 24) on the order of 100 bytes each. The record consists of two numbers (Surface Radiance and Top of Atmosphere Radiance) for each of the possible values of reflectance (i.e. the number of values in a LUT record is two times the number of reflectances used). The LUT record will be used in an interpolation scheme to obtain the proper correction.

The ASTER VNIR/SWIR atmospheric correction requires a table estimated at 10 Gigabytes in size. The table can be split by ASTER band number into 9 sub-tables since the bands are handled independently (each sub-table being approximately 1 Gigabyte). If the investigator only wants SWIR, the number of sub-tables (in a LUT) would be 6 while for VNIR, there would be 3 or 4.

The look-ups (queries if a database is used) will be based on seven parameters for each band and file version. (See 3 through 9 in the list above).

Access to a LUT record is based on combinations of 8 parameters. The first parameter is band number, and the remaining 7 parameters are derived from values which characterize the atmosphere.

There are 9 bands. The remaining 7 parameters are expected to assume approximately 10 values. So, an estimate of the size of the table is:

9 X 10E7 X 100 bytes = about 9 - 10 Gigabytes

Since the 7 other parameters are floating point values, they cannot be used directly as indices or keys to the table. Seven auxiliary tables will be used to find the integer key corresponding to an input floating point value.

The number of lookups per scene depends on the resolution of the chosen input data. The resolution may be different for MISR, MODIS and NMC (e.g., if MODIS data is available, the grid may be tighter than if only NMC data is available.) It is currently estimated that for MISR data, accesses per scene will range from 200 to 40,000 with an average number of 1,500 accesses per scene. With a processing load of 70 scenes per day, this implies an average access rate of 1 access every second assuming a distributed processing load.

It is expected that several versions of the LUT may be delivered over the life of the project. These additional versions will not be just updates (due to instrument changes) or improvements to the preceding versions, but may account for special atmospheric conditions such as volcanic eruptions. So, it could well be the case (especially if the ASTER Surface Radiance Product is virtually archived) that two or more tables would need to be staged during one day.

It is not known whether these tables will grow or shrink in size over the years. Some could become denser while others may be trimmed.

6.1.4.3 Database vs. Flat File ASTER Lookup Table

The large number of accesses (several hundred per scene) required by the ASTER software to the ASTER LUT rules out the possibility of allowing this software access to the data server (either directly or via the SDP toolkit) during execution since this will contest with other users and other data production activities for data server resources and query satisfaction. This leaves open the option of either implementing the LUT as a set of files that could be staged or incorporating the LUT into a separate database that could be directly accessed by the ASTER SW during execution.

The option of storing the LUT as a separate data type in the data server (i.e. an instantiated data server) was considered and rejected. It would be difficult to justify the significant cost associated with developing an instantiated data server that is relevant only to ASTER internal processing and would not be required by any other external user. Considerable cost is associated with the effort needed to establish a new data type [in the data server], define new, related services and ensure that ASTER data access performance requirements are met. In addition, by storing the LUTs in a data sever, ASTER data access and retrieval performance would be adversely impacted . Use of a data server introduces additional layers of supporting data that must be traversed before the actual data to be retrieved can be reached. (In addition, the Data Server design needs to be mature by Jan. 1996 (or earlier).) This unnecessary complexity would be eliminated if the ASTER software has direct access to a separate independent database storing the LUTs.

(A) Characteristics of a LUT Database

A 10 GB LUT would require approximately 20 -25 GB of disk space if stored in a relational DBMS. The additional storage is required for the secondary indices used to accelerate searching. A standard commercial relational DBSM could be used to provide the required functionality and the necessary performance. Additional characteristics are as follows:

- Requires software support to provide the capability to create, populate, update, and delete database relations or tables. Defining a relation means specifying a list of fields, each with a specifiable type.

The minimum set of types the database should support are:

- integer
- floating point (or real)
- character strings
- date/time (which the database would have to know how compare and convert)
- image (binary data - up to sizes on the order of 32K)

- Requires capability to allow software program to query database tables and manipulate the results in program variables (prefer SQL). The query facility should support, at a minimum, expressions employing logical ‘and’ and ‘or’. Join's would be an added plus - the ability to query more than one table and pick a record from one based on some condition in the other, and so on.
- A relational database (such as SYBASE) would be preferred owing to the nature of the queries involving 7 parameters.
- The database needs backup capability, security features and accompanying statistical evaluation tools.
- Efficient (optimized) database server is needed to make efficient use of the disk controllers.

(B) PROS and CONS of using a Database

PROS:

1. Automatically provides simultaneous access [to the LUTS] from any machine running ASTER science software. Remote mounting of file systems from different processors would not be necessary.
2. Commercial databases are designed to handle storage, update, and retrieval of large volumes of data. It also provides security, backup and statistical evaluation tools for the database.
3. Easy to implement the ASTER software. Allows easy and efficient access to LUT records using simple queries containing the 7 parameters. Would allow ASTER software to proceed with their implementation and meet their schedules (I&T, Jan. 96)
4. Maintenance and update of LUT tables easier via database tools vs. maintaining large files.
5. Could be used by additional ASTER special product software which would ease the transition of those products to standard products if so required.
6. A new version of a LUT table (involving a small change to an existing LUT) can be created by relinking existing columns to new ones thereby requiring only a small increase in disk space (i.e. to support a change to, say, one band of a telescope or increase resolution in just a portion of the table). A new LUT in a file system would always require 10 GB additional disk space.
7. Since ASTER LUTs are needed only by the ASTER software (and are internal to its processing), setting up a separate, independent database to support storage and retrieval of

ASTER LUTS seems logical. ASTER software could then have control over how they set up their interface with this database, (via vendor-supplied APIs) and can ensure that they meet their data access performance requirements.

CONS:

1. Additional cost associated with the SDP toolkit to support interface of ASTER software to database. (*Response*: ASTER science team would like a waiver from the use of the SDP toolkit. Could use a vendor-supplied API to the database.)
2. Additional cost associated with the setup and population of a database from the set of files making up the original look-up table. (*Response*: ASTER science team indicated that they would support setup and population of the database. However licensing cost and cost of future maintenance of database must be considered.)
3. Performance impacts related to relational database access and retrieval depend on degree of optimization of database server access to disk controller and networks vs. the operating system file system access routines used. (*Response*: While the relational DBMS requires more overhead than the file system access routines, the ASTER performance requirements are well within reach of current commercial relational DBMS technology. The ASTER team will be responsible for designing the database schema / queries and ensuring that data access performance requirements are met.)

NOTE. It is currently difficult to assess performance impacts of using a database vs. flat files without some sort of prototype. However the cost of developing a prototype is prohibitive (comparable to that of developing the actual software).

4. An entirely new LUT would require 10 Gbytes using the file system while an additional LUT would need 20 GB if stored in the database. However, ASTER science team indicated that updates could be accommodated by replacing one LUT with another. The maximum we would have to size for to accommodate updates (patches, added columns) would be about 40 - 50 Gbytes.
5. Any future upgrades / changes to the database would fall within the ASTER project's realm of responsibility. (ASTER science team does not (at present) foresee any need to upgrade or change their choice of database once selected.)
6. Potential processing performance impact if CPU is idle while waiting for query completion. However, same or worse impact might result from flat files.

The average database query response time is not currently known since a prototype has not yet been developed. However, during a query, chances are the processor will proceed with related processing and will not be idle.

(C) Specifics of using a LUT Flat File System

This approach would entail breaking the LUT into a set of files based on bands and developing ASTER software to access the appropriate file after calculating the values of the parameters. The ECS science processing environment would pre-stage all files prior to the execution of the ASTER software.

(D) PROS and CONS of a Flat File System

PROS:

1. LUT handled like any other input data, i.e. it can be pre-staged.
2. No cost to the SDP Toolkit.
3. Would use less disk storage than the DBMS approach. (An additional LUT would require 10 GB using the file system while an additional LUT could require additional 20 GB if stored in the relational database.)

CONS:

1. Involves large number of files to be staged. The entire LUT (10 gigabytes) would have to be pre-staged since the specific table lookups are not known at staging time. Since different versions may be required, potentially more than 10 gigabytes may need to be staged during the day.
2. The Process Control Information (PC_INFO) file (used by the SDP toolkit for file identification) may have difficulty maintaining the numerous entries corresponding to the LUT files. The size of each file would be limited by the maximum file size allowed by the operating system.
3. More complex coding required of ASTER SW to efficiently partition, access and retrieve data from the LUT files. Greater cost associated with generating ASTER software to manipulate files (involving more FLOPS). This can degrade speed of execution of ASTER software. Also portability of these files is required to support the different types of machines.
4. Software integration and test of updated flat files would be considerable. Each file could contain potentially 1,000,000 records of 100 bytes each. Maintenance of files may be limited to replacing or adding whole files since updating extremely large files could be extremely cumbersome.

6.1.4.4 Conclusions

After studying the pros / cons of implementing ASTER LUTs as either a database system or a flat file system and after considering other selection criteria (such as performance impacts, cost, risk in implementing, stability of design, maintainability and evolvability), the database approach has been selected. The ASTER LUTs would be stored in a dedicated database at the processing site (EDC) from where they can be directly accessed by the ASTER science software during execution.

A relational database (such as SYBASE / SQL) is recommended over an Object-Oriented Database (OODBMS) since complex data objects are not used. Also, it is not easy to interface a C or FORTAN program directly to an OODBMS. (Since ASTER software is in the C and FORTRAN programming languages, a C++ interface would have to be developed if an OODBMS were chosen. Using an independent relational database implies that a vendor-supplied C or FORTRAN API can be used. ASTER science team would like a waiver from using the SDP Toolkit interface.)

The characteristics of the LUT database have been presented in Section 6.1.4.3. Initially only one LUT (25 Gbytes maximum) is required to be stored in the database. Over time this could be replaced with a newer (updated) version or updated with patches (e.g. added columns). The database could grow to an estimated maximum of 40 - 50 Gbytes. Database administration activities would fall under Processing Operations.

The hardware required to support the LUT database consists of two workstation class hosts (with 25 Gbytes storage each) to hold the DBMS server.

The cost of the flat file approach has been presented below along with the cost of the DBMS approach for purposes of comparison.

6.1.4.5 Costs

The sharing of cost associated with setting up a separate database for the LUTs is something that needs to be worked out between the ESDIS, SDPS and ASTER teams.

The following should be costed when estimating the cost of storing the LUTs in a separate database:

- Estimated hardware cost associated with 2 work-station class hosts for the DBMS server.
- Estimated cost to license a (relational) database (for example: SYBASE) and support software (OPEN client, SQL server and SQL monitor).
- Estimated cost of database service contract (till year 2006).
- Estimated cost of associated database administration functions (from launch) till year 2006.

The following should be costed when estimating the cost of storing the LUTs in flat files (for purposes of comparison):

- Estimated hardware cost to store 10 Gbytes worth of data on local disk storage.
- Estimated ASTER software cost to develop and maintain software related to file data management functions on the LUTs stored in the flat files (from present till 2 years after launch (June, 1998) and then 6 years beyond that (till year 2006).
- Administration costs.

7. Recommendations and Conclusions

As a result of this trade study, the following recommendations and conclusions have been drawn concerning the direct data access requirements (during execution) of the science software and how they will be supported:

ASTER:

It has been determined that ASTER science software has a definite need to access its lookup table (LUTs) during execution. To support this need, the option of allowing these LUTs to reside in a separate database, (dedicated to ASTER processing) was chosen after assessing the pros / cons and cost of each option. (The cost of storing the LUTS in a separate database vs. flat files has been estimated and the separate database option was found to be more cost-effective.) To minimize overall cost, it is further recommended that ASTER be given an SDP Toolkit waiver so that the DBMS may be accessed directly from the science software.

The sharing of cost associated with setting up a separate database for the LUTs is something that needs to be worked out between the ESDIS, SDPS and ASTER teams.

Other Instruments:

To date no direct access data requirements (during execution) have been identified for CERES or MISR. MODIS direct access data needs are still being studied and will be known after the study on MODIS processing scenarios is complete (around the CDR timeframe).

Plan to continue working with the algorithm/science software development teams in determining whether any additional direct data access requirements (during execution) need to be identified.

This page intentionally left blank.

Abbreviations and Acronyms

API	Application Programming Interface
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer (formerly ITIR)
CADM	CERES Angular Directional Models
CDR	Critical Design Review
CERES	Clouds and Earth's Radiant Energy System
DAAC	Distributed Active Archive Center
DBA	Database Administration
DBMS	Database Management System
ECS	EOSDIS Core System
EOSDIS	Earth Observing System Data and Information System
ESDIS	Earth Science Data and Information System
GB	Gigabytes
LUT	Lookup Table
MISR	Multi-Angle Imaging Spectroradiometer
MODIS	Moderate-Resolution Imaging Spectroradiometer
OODBMS	Object-Oriented Database Management System
PDPS	Planning and Processing System
PGS	Product Generation Subsystem
SDPS	Science Data Processing Segment
TRMM	Tropical Rainfall Measuring Mission